DATABASE

ชัยยงค์ วงศ์ชัยสุวัฒน์

การเชื่อมต่อและการเรียกใช้ฐานข้อมูลหลายแหล่ง

กล่าวถึงสถาปัตยกรรมการเชื่อมต่อ และการเรียกใช้ฐานข้อมูลหลายแหล่ง และกล่าวถึงที่มาของ ODBC

ระบบฐานข้อมูลภายในองค์กรส่วนใหญ่มักจะเป็นฐานข้อมูลแบบ heterogeneous database ที่มาจากหลายตระกูลตามแต่การจัดเก็บใช้งานของแต่ละหน่วยงานย่อยขององค์กรนั้น บางหน่วยงานย่อยอาจจะเก็บข้อมูลบนเครื่องพีซีด้วยการใช้ซอฟต์แวร์จัดการฐานข้อมูลขนาดเล็กตระกูล XBase เช่น ดีเบส ฟอกซ์โปร์ หรือคลิปเปอร์ บางแห่งอาจใช้ของค่าย Borland คือ พาราด็อกซ์ บางแห่งก็อาจใช้เก็บข้อมูลเป็นไฟล์สเปรดชีตด้วยโลตัส 123 หรือเอ็กเซล หรือบางแห่งเก็บข้อมูลรายการย่อยด้วยฐานข้อมูล Btrieve บนระบบเครือข่ายของ Novell Netware

ในกรณีของหน่วยงานใหญ่ขึ้นก็จำเป็นต้องใช้เก็บข้อมูลขนาดใหญ่ปริมาณมาก ๆ ด้วยซอฟต์แวร์จัดการฐานข้อมูลดีขึ้น โดยมีระบบความถูกต้อง ความปลอดภัยของการใช้ข้อมูลที่เชื่อถือวางใจได้ เครื่องคอมพิวเตอร์ที่ใช้เก็บข้อมูลก็ควรเป็นระบบที่ใหญ่ขึ้น ขีดความสามารถสูงขึ้น เช่นเครื่องเวอร์ก-สเตชัน หรือเครื่องพีซีขนาดใหญ่ เพื่อมาทำเป็นเครื่องผู้ให้บริการเซอร์ฟเวอร์ในสถาปัตยกรรมแบบ Client/Server ที่กำลังได้รับความนิยมอย่างมากในปัจจุบัน ฐานข้อมูลในกลุ่มนี้ได้แก่ Oracle, INFORMIX, INGRES, SQL Server, SQLBase, Netware SQL

เป็นต้น

บางหน่วยงานก็อาจมีการใช้เก็บข้อมูลด้วยซอฟต์แวร์จัดการฐานข้อมูลเฉพาะของเครื่องบนเครื่องมินิคอมพิวเตอร์และเมนเฟรม ในเมื่อข้อมูลเก็บไว้หลายแหล่งหลายตระกูลเช่นนี้ การเชื่อมต่อและเรียกใช้จึงค่อนข้างจะยุ่งยากสับสนกับผู้ใช้งานมากตลอดอดีตที่ผ่านมา แต่ในปัจจุบันการเรียกค้นข้อมูลจากหลายแหล่งหลายตระกูลเพื่อมาประกอบในโปรแกรมประยุกต์เริ่มดูสดใส และมีลู่ทางทำได้อย่างง่ายดายด้วยซอฟต์แวร์เข้ามาแทรกเสริมระหว่างผู้รับบริการ Client กับผู้ให้บริการเซอร์ฟเวอร์

บทความนี้จะกล่าวถึงสถาปัตยกรรมการเชื่อมต่อและเรียกใช้ฐานข้อมูลหลายแหล่งซึ่งมีอยู่สามแบบใหญ่ ๆ และจะเน้นที่แบบที่หนึ่งซึ่งเป็นแบบอินเตอร์เฟสร่วม โดยขณะนี้ได้มีการกำหนดมาตรฐานของการอินเตอร์เฟสหรือเชื่อมต่อไว้แล้ว บริษัทไมโครซอฟต์ก็ได้นำเกณฑ์มาตรฐานนั้นมากำหนดฟังก์ชันร่วมสำหรับการเรียกใช้ฐานข้อมูลหลายแหล่งหลายตระกูลที่เรียกว่า Open Database Connectivity นิยมเรียกย่อ ๆ ว่า ODBC ในบทความนี้จะกล่าวที่มาของ ODBC พร้อมกับตัวอย่างซอฟต์แวร์ ODBC Driver ของบริษัท Q+E Software, Inc. ซึ่งสามารถเลือกติดต่อกับแหล่งฐานข้อมูลมากกว่า 20 แบบ

แนวโน้มระบบคอมพิวเตอร์

เทคโนโลยีคอมพิวเตอร์มีการปรับเปลี่ยนอย่างรวดเร็วอย่างไม่หยุดนิ่ง ปัจจุบันเครื่องแบบตั้งโต๊ะราคาถูกแต่ขีดความสามารถสูงเข้ามามีบทบาทแทนที่เครื่องรุ่นเก่า ซึ่งถ้านำมาต่อเข้าเป็นระบบเครือข่ายด้วยก็ยังเสริมขีดความสามารถให้เด่นยิ่งขึ้น จนมาเบียดแทรกแทนเครื่องมินิคอมพิวเตอร์และเมนเฟรม หน่วยงานต่าง ๆ มีการลดขนาดปรับเปลี่ยนไปใช้เครื่องที่เล็กลงให้พอเหมาะกับงานจริง ๆ ตรงกับคำศัพท์ที่นิยมใช้กันมากในปัจจุบันคือ Down-Sizing และ RightSizing นั่นเอง

แนวโน้มการใช้งานคอมพิวเตอร์กำลังก้าวหน้าจากยุคของ End-User Computing ที่ผู้ใช้งานแต่ละคนต่างมีเครื่องพีซีทำงานแบบ Stand alone กันมากมาย ซอฟต์แวร์ประยุกต์ที่ใช้ก็เป็นซอฟต์แวร์หลัก ๆ โปรแกรมของเวิร์ดโปรเซสเซอร์ สเปรดชีต การจัดการฐานข้อมูล การนำเสนอข้อมูลภาพกราฟิก และการติดต่อสื่อสาร ไปสู่ยุคที่เรียกว่า Workgroup Computing ที่ประกอบด้วยกลุ่มผู้ใช้หลายคน ทำงานร่วมกันบนระบบคอมพิวเตอร์เชื่อมต่อเป็นเครือข่ายคอมพิวเตอร์ เป็นการใช้ข้อมูลและทรัพยากรในระบบร่วมกัน ระบบแบบ Workgroup Computing ไม่ได้เข้ามาแทนที่ระบบ End-User Computing เพียงแต่แนวโน้มการใช้งานแบบผู้ใช้เดี่ยว ลักษณะ Stand alone จะลดน้อยลง

เทคโนโลยีการสื่อสารข้อมูลได้พัฒนารุดหน้าอย่างรวดเร็ว ทำให้แนวโน้มการใช้ประโยชน์จากเครือข่ายเพิ่มสูงขึ้น การแลกเปลี่ยนติดต่อข้อมูลในระยะไกลสามารถติดต่อถึงกันในเวลารวดเร็วถูกต้องปลอดภัย การใช้งานประยุกต์ต่าง ๆ สามารถทำงานอยู่กับเครื่องรับบริการ หรือ Client ที่เป็นเครื่องแบบตั้งโต๊ะ ด้วยการส่งงานติดต่อกับเครื่องให้บริการเซอร์ฟเวอร์ที่อยู่ห่างไกลอย่างสะดวกสบาย ซึ่งในกรณีที่

นิยมใช้ปัจจุบันเครื่องให้บริการเซอร์ฟเวอร์จะเป็นส่วนที่เก็บฐานข้อมูลหลัก ดังนั้นบางครั้งเรานิยมเรียกว่า Database Server หรือ Database Engine

Fedarated Database

การจัดการฐานข้อมูลภายในองค์กรที่มีการกระจาย ในสมัยก่อนเราเคยคิดว่าต้องเป็นฐานข้อมูลแบบกระจาย (Distributed Database) โดยมองฐานข้อมูลที่กระจายเก็บหลาย ๆ แหล่งในทางกายภาพเสมือนเป็นฐานข้อมูลเดี่ยว หากเป็นฐานข้อมูลแบบกระจายจริง ๆ ระบบจัดการฐานข้อมูลในทุก ๆ แหล่งเก็บก็ต้องเป็นระบบจัดการเดียวกัน และใช้โครงสร้างข้อมูลโมเดลแบบเดียวกัน ฐานข้อมูลเดิมที่เคยจัดเก็บแล้วก็ต้องมีการแปลงให้อยู่ในรูปแบบของระบบจัดการฐานข้อมูลใหม่ ดังนั้นจึงดูค่อนข้างยุ่งยากพอสมควร

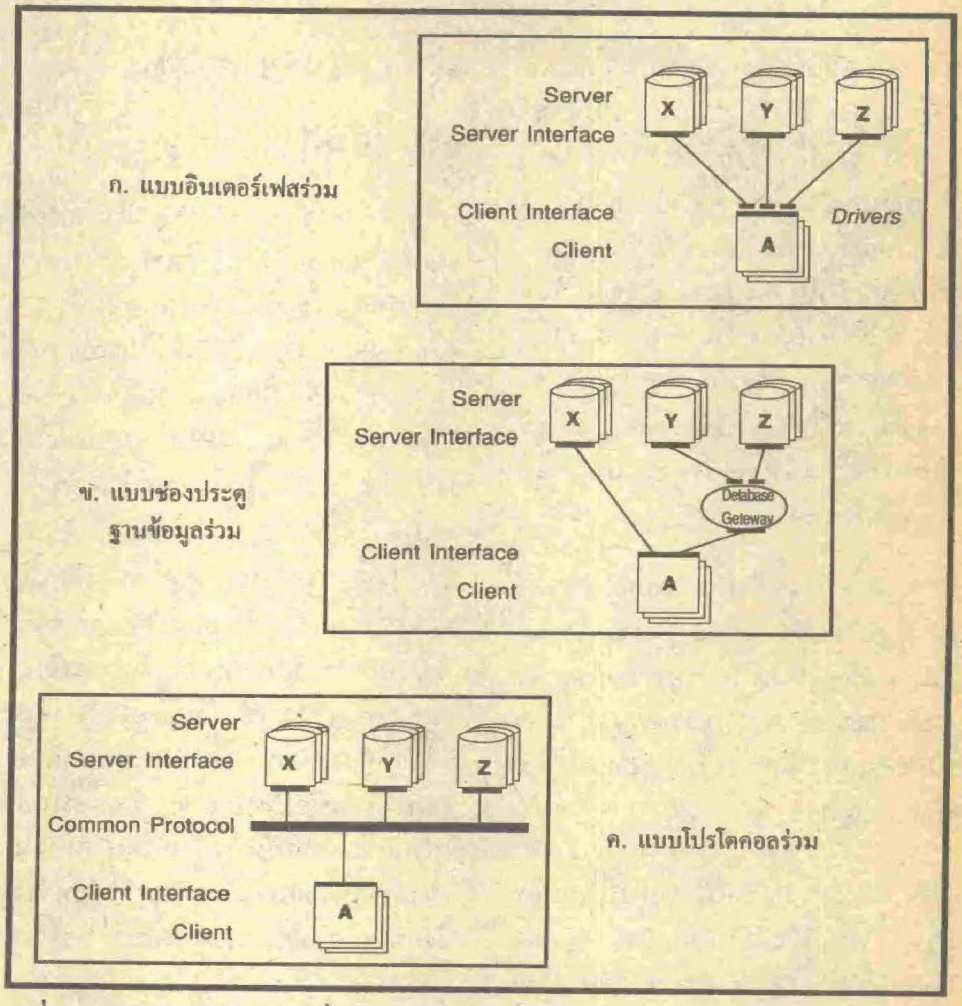
ฐานข้อมูลภายในองค์กรซึ่งมีการจัดเก็บอยู่แล้วด้วยระบบจัดการฐานข้อมูลหลายตระกูล หากสามารถเชื่อมโยงติดต่อเรียกใช้จากหลายแหล่งมาใช้ประโยชน์ร่วมกันโดยการจัดการดูแลในด้านความถูกต้องข้อมูล ระบบความปลอดภัยข้อมูล การใช้งานร่วมกัน และการบำรุงรักษาร่วมกันอย่างเป็นระบบ

ในลักษณะเช่นนี้จะไม่เรียกว่าฐานข้อมูลกระจาย แต่อาจเรียกว่าฐานข้อมูลสหพันธ์ หรือฐานข้อมูลร่วมกัน (Fedarated Database)

ปัญหาสำคัญของฐานข้อมูลสหพันธ์ของการเรียกใช้ข้อมูลหลายแหล่งร่วมกัน จะอยู่ที่การกำหนดโครงสร้างหรือสถาปัตยกรรมเชื่อมต่อ ให้ผู้ใช้ที่เครื่องรับบริการ หรือ Client สามารถติดต่อเรียกใช้ข้อมูลจากแหล่งฐานข้อมูลที่เครื่องให้บริการเซอร์ฟเวอร์โดยไม่ต้องคำนึงถึงความหลากหลายของเครื่องเซอร์ฟเวอร์ว่าจะมีโครงสร้างอย่างไร และจะใช้ภาษาเรียกติดต่ออย่างไร

สถาปัตยกรรมเชื่อมต่อ

สถาปัตยกรรมการเชื่อมต่อจะเป็นวิธีการหรือรูปแบบที่อำนวยความสะดวกในการเรียกใช้ข้อมูลแหล่งต่าง ๆ แนวทางสถาปัตยกรรมมีอยู่ 3 แบบ (ดังรูปที่ 1) คือ



รูปที่ 1 แนวทางสถาปัตยกรรมเชื่อมต่อ

ก. แบบอินเตอร์เฟสร่วม (Common Interface)

วิธีการนี้เป็นการเชื่อมต่อจากโปรแกรมประยุกต์ใช้บนเครื่อง Client เพื่อติดต่อกับแหล่งข้อมูลที่เซอร์ฟเวอร์ โดยโปรแกรมประยุกต์นี้เพียงออกแบบและสร้างขึ้นครั้งเดียว แต่สามารถใช้กับข้อมูลหลายแหล่งหลายตระกูล ซึ่งโปรแกรมประยุกต์จะเรียกใช้ข้อมูลจากแหล่งข้อมูลภายนอกผ่านทางโปรแกรมไดรเวอร์เฉพาะฐานข้อมูลตระกูลหนึ่ง ๆ หากต้องการเปลี่ยนแปลงแหล่งข้อมูลตระกูลอื่น ก็เพียงเปลี่ยนโปรแกรมไดรเวอร์ของการเชื่อมต่อเท่านั้น โปรแกรมประยุกต์ไม่ต้องแก้ไข หรือหากจำเป็นต้องแก้ไขก็เพียงเล็กน้อยมาก

ข. แบบใช้ช่องประตูร่วม (Common gateway)

วิธีนี้เป็นการใช้ซอฟต์แวร์ที่เปรียบเสมือนเป็นช่องประตูสำหรับการติดต่อระหว่างผู้รับบริการ Client ไปยังผู้ให้บริการเซอร์ฟเวอร์ ซอฟต์แวร์ช่องประตูนี้จะช่วยในการแปลการร้องขอจาก Client ส่งไปยังเซอร์ฟเวอร์ ขณะเดียวกันจะเป็นตัวแปลผลลัพธ์ข้อมูลที่ได้คืนจากแหล่งข้อมูลเซอร์ฟเวอร์ไปยัง Client อีกทีหนึ่ง วิธีนี้โปรแกรมประยุกต์บนเครื่องผู้รับบริการ Client ที่เคยติดต่อกับแหล่งข้อมูลตระกูลหนึ่งก็สามารถเรียกติดต่อตระกูลอื่นผ่านช่องประตูฐานข้อมูลอื่นด้วย ซึ่งฐานข้อมูลอื่นนี้มักเป็นฐานข้อมูลที่อยู่ในเครื่องต่างระดับกัน โดยทั่วไปนิยมติดต่อกับฐานข้อมูลบนเครื่องเมนเฟรม

ค. แบบใช้รูปแบบโปรโตคอลร่วม (Common Protocol)

วิธีนี้เป็นการกำหนดรูปแบบร่วมที่ได้รับการออกแบบอย่างดีสำหรับการต่อเชื่อมระหว่างโปรแกรมประยุกต์ที่ Client กับแหล่งข้อมูลของเซอร์ฟเวอร์หลายตระกูล ณ จุดต่อเชื่อมของ Client จะเป็นซอฟต์แวร์รับการร้องขอจากโปรแกรมประยุกต์ Client แล้วแปลง

การร้องขอให้เข้ารูปแบบร่วม (Common Protocol) ขณะเดียวกันทางฝ่ายของผู้ให้บริการเซอร์ฟเวอร์จะรับการร้องขอในรูปแบบร่วม เรียกหาข้อมูลที่ต้องการแล้วแปลงผลลัพธ์ข้อมูลให้อยู่ในรูปแบบร่วมคืนไปให้ผู้รับบริการ Client วิธีนี้มีข้อดีที่เหมาะสำหรับการเพิ่มประเภทของ Client ใหม่และประเภทของเซอร์ฟเวอร์ใหม่เข้ามาภายหลัง เพราะเป็นเพียงการแปลงเข้าหารูปแบบร่วมที่มีการกำหนดไว้

ตัวอย่างผลิตภัณฑ์แบบอินเตอร์เฟสร่วมได้แก่ Apple Data Access Language (DAL), DBI Enterprise Data Access/SQL (EDA/SQL) และ Micorsoft Open Database Connectivity (ODBC)

ตัวอย่างผลิตภัณฑ์แบบใช้ช่องประตูร่วม ได้แก่ IBM Distributed Database Connection Service (DDCS/2), Microsoft Open Data Service (ODS), Sybase Open Server และ MDI Database Gateway for DB/2

ตัวอย่างผลิตภัณฑ์ที่ใช้รูปแบบโปรโตคอลร่วมได้แก่ Sybase Tabular Data Stream (TDS), IBM Distributed Relational Database Architecture (DRDA) และ ISO Remote Database (RDA)

ในที่นี้จะแยกอธิบายเฉพาะแบบอินเตอร์เฟสร่วมในส่วนของผลิตภัณฑ์ของ Microsoft ODBC ซึ่งกำลังได้รับความนิยมนำมาใช้กับซอฟต์แวร์หลัก ๆ ของไมโครซอฟต์เอง

SQL ภาษามาตรฐานฐานข้อมูล

SQL มาจากคำว่า Structured Query Language ที่เป็นภาษาใช้ในการเรียกค้นข้อมูล ซึ่งมีจุดกำเนิดดั้งเดิมมาจาก SEQUEL ที่เป็นส่วนหนึ่งของโครงการวิจัย System R ของห้องปฏิบัติการวิจัยของ IBM ที่ SanJose ราว ค.ศ. 1974 และมีการนำมาใช้ในซอฟต์แวร์จัดการฐานข้อมูล ORACLE และ SQL/DS บนโปรแกรมปฏิบัติการ VM/370 ของ IBM ต่อมาได้ขยายมาใช้ใน DB/2 บนโปรแกรมปฏิบัติการ MVS/370 ของ IBM เช่นกัน ภาษาเรียกค้นนี้ได้รับความนิยมนำไปใช้ในซอฟต์แวร์จัดการฐานข้อมูลใหญ่ ๆ อีกแทบทุกโปรแกรม จนมีการกำหนดมาตรฐานของภาษานี้โดยสถาบันมาตรฐานระหว่างประเทศ (ISO) และสถาบันมาตรฐานสหรัฐอเมริกา (ANSI)

มาตรฐานภาษา SQL ออกมาครั้งแรก ค.ศ. 1986 และปรับปรุงขยายใน ค.ศ. 1989 ในระหว่าง ค.ศ. 1992 ก็ได้เพิ่มขยายมาตรฐานขึ้นกลายเป็น SQL-2 หรือ SQL 92 ตามปีที่กำหนดมาตรฐาน ขณะนี้กำลังร่างมาตรฐานภาษา SQL รุ่นที่ 3 เป็น SQL 3 อยู่

ในการเชื่อมต่อและเรียกใช้ฐานข้อมูลหลายแหล่ง ภาษา SQL ได้รับการเลือกแล้วให้ถือเป็นภาษากลางสำหรับการเรียกติดต่อฐานข้อมูลด้วย

SQL Access Group (SAG)

SAG เป็นหน่วยงานหนึ่งที่มีหน้าที่ชักนำให้ผู้ผลิตระบบและซอฟต์แวร์ต่าง ๆ มากำหนดรูปแบบร่วมสำหรับฐานข้อมูลในการโอนย้ายและทำงานร่วมกันได้ ด้วยการสนับสนุนมาตรฐาน SQL หน่วยงานนี้ก่อตั้งเมื่อ ค.ศ. 1989 โดยเป็นสถาบันไม่แสวงหาผลกำไร สมาชิกส่วนใหญ่จะเป็นผู้อยู่ในวงการฐานข้อมูล ได้แก่ DEC, HP, Informix, Ingres, Oracle, Sun, Tandem, Teradata เป็นต้น

บทบาทของ SAG ในระยะที่ผ่านมาคือการร่างข้อกำหนดร่วมของภาษาเรียกค้น ANSI SQL สำหรับการเชื่อมต่อด้านผู้รับบริการ Client และการร่างข้อกำหนดร่วมของโปรโตคอลที่เรียก RDA (Remote Database Access) ซึ่งเป็นการเชื่อมต่อและเรียกใช้ข้อมูลแบบที่สามที่เคยกล่าวไว้แล้ว นับเป็นงานที่มาสนับสนุนหรือเสริมสถาบัน ISO ในการสร้างข้อกำหนดร่วมที่ยังไม่กระจ่างให้ชัดเจนยิ่งขึ้น ข้อกำหนดร่วมต่าง ๆ นี้ยังได้รับการทบทวนตรวจสอบจากบริษัท X/open ที่ทำงานด้านการโอนย้ายร่วมกันได้ของโปรแกรมคอมพิวเตอร์ต่าง ๆ ระหว่างเครื่องต่างรุ่นต่างระดับกัน เป็นการผลักดันให้เกิดระบบเปิดที่แท้จริงต่อไปในอนาคต

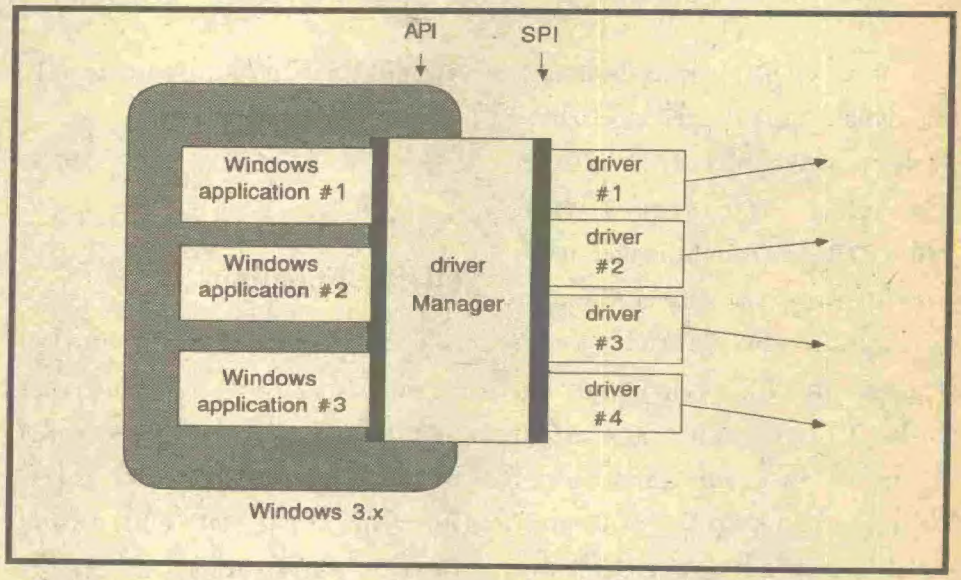
งานส่วนที่สามที่หน่วยงาน SAG มีบทบาทคือการออกข้อกำหนดร่วมของ SQL Call-Level Interface (SQL CLI หรือ CLI) คือเป็นฟังก์ชันที่สามารถเรียกใช้ได้โดยตรงจากโปรแกรมประยุกต์ ในรูปของ API (Application Programming Interface) ซึ่งวิธีนี้ช่วยให้เรียกใช้ภาษา SQL ได้ง่ายขึ้น ไม่ต้องเรียกใช้ด้วยการฝังส่วนของภาษา SQL รวมในภาษาโปรแกรมมิงอื่น ๆ ในลักษณะ embedded SQL นับ

เป็นส่วนของเครื่องมือเพื่อให้นักพัฒนาโปรแกรมใช้ในการเชื่อมต่อเรียกใช้ข้อมูลหลายแหล่งหลายตระกูล จากโปรแกรมที่ Client โดยไม่ต้องดัดแปลงแก้ไข เพียงแต่เปลี่ยนไดรเวอร์เชื่อมต่อตามสถาปัตยกรรมเชื่อมต่อแบบอินเตอร์เฟสร่วมนั่นเอง

Windows Open Service Architecture (WOSA)

บริษัทไมโครซอฟต์ผู้เป็นซอฟต์แวร์เฮ้าส์ใหญ่ที่สุดของโลก เจ้าของเอ็มเอสดอส วินโดวส์ ซอฟต์แวร์หลักในทางธุรกิจใหญ่คือ เวิร์ด เอ็กเซล แอคเซสเพาเวอร์พอยน์ เมล์ โปรเจ็กต์ ฯลฯ และซอฟต์แวร์ตัวแปลภาษาต่าง ๆ เช่น ฟอร์แทรน ซี ซีพลัสพลัส โคบอล เบสิก และอื่น ๆ อีกมากมาย ได้พยายามโปรโมตผลิตภัณฑ์โดยเฉพาะวินโดวส์อย่างเต็มที่ จึงได้กำหนดข้อกำหนดเพื่อการบริการร่วม ซึ่งบริษัทซอฟต์แวร์อื่น ๆ สามารถนำไปใช้ในโปรแกรมประยุกต์ของงานเพื่อมาเสริมใช้งานวินโดวส์กว้างขวางยิ่งขึ้น ข้อกำหนดบริการดังกล่าวได้แก่ การเชื่อมต่อเรียกใช้ข้อมูล (ODBC) การใช้งานหลายสื่อหรือมัลติมีเดีย (multimedia) และการส่งข้อความทางอิเล็กทรอนิกส์ หรืออิเล็กทรอนิกส์ไปรษณีย์ (E-mail)

รูปที่ 2 แสดงสถาปัตยกรรมของ WOSA ที่โปรแกรมประยุกต์บนวินโดวส์ติดต่อไปยังส่วนจัดการไดรเวอร์ผ่าน API และจากนั้นจะผ่าน SPI ไปยังไดรเวอร์ต่าง ๆ ไปยังแหล่งข้อมูลอื่น ๆ

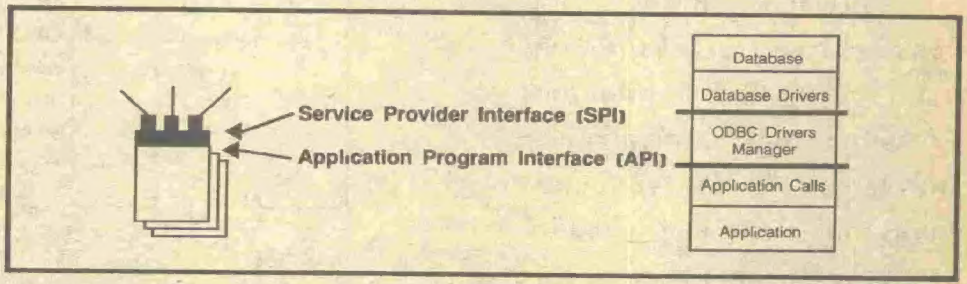


รูปที่ 2 Windows Open Service Architecture ของไมโครซอฟต์

Microsoft Open Database Connectivity (ODBC)

ODBC เป็นการเชื่อมต่อและเรียกใช้ข้อมูลแบบการอินเตอร์เฟสร่วมที่ประกอบด้วยส่วนของ generic interface ที่เรียก ODBC Driver Manager และส่วนของไดรเวอร์สำหรับเชื่อมต่อกับแหล่งฐานข้อมูล ODBC นี้ออกแบบสำหรับไมโครซอฟต์วินโดวส์และวินโดวส์เอ็นที สำหรับโปรแกรมภาษาซีในการใช้เรียกระดับ call-level ขณะนี้กำลังจะขยายนำไปใช้กับแอปเปิ้ลแมคอินทอช แต่ไม่ได้ใช้ในดอส หรือโอเอสทู นับเป็นความพยายามอีกครั้งหนึ่งของไมโครซอฟต์ที่หวังให้โปรแกรมประยุกต์บนเครื่อง Client ขึ้นไปสู่ระดับวินโดวส์ของตนเองมากที่สุด

โครงสร้างของ ODBC กับการอินเตอร์เฟสร่วมฐานข้อมูลแหล่งต่าง ๆ อาจแสดงได้ดังรูปที่ 3 โดยโปรแกรมประยุกต์ในรูปของ Application Call จะติดต่อกับ generic interface ซึ่งก็คือ ODBC Driver Manager ผ่านทาง Application Programming Interface (API) ในส่วนติดต่อกับฐานข้อมูลก็เป็นการเชื่อมต่อจาก ODBC Driver Manager ผ่าน Service Provider Interfacd (SPI) ไปยังไดรเวอร์เพื่อส่งไปยังฐานข้อมูลแหล่งต่าง ๆ ซึ่ง Service Provider Interface นี้เป็นข้อกำหนดที่เปิดกว้างให้บริการต่าง ๆ สร้างหรือผลิตไดรเวอร์โดยไม่ขึ้นกับโปรแกรมประยุกต์และ Driver Manager ผู้ใช้งานสามารถเลือกซื้อผลิตภัณฑ์ที่ยึดข้อกำหนดมาใช้ติดต่อเรียกใช้ฐานข้อมูลต่าง ๆ มากมาย



รูปที่ 3 โครงสร้าง ODBC ในรูปแบบอินเตอร์เฟสร่วม

ส่วน ODBC Driver Manager เป็น dynamiclink library ของไมโครซอฟต์เอง ถ้าเราติดตั้งโปรแกรม Excel หรือ Access ของไมโครซอฟต์ก็จะได้ไฟล์ ODBC Driver Manager มาให้ด้วย หน้าที่หลักของ Driver Manager จะใช้ในการโหลดไดรเวอร์ ซึ่งเราสามารถกำหนดให้ติดตั้งไดรเวอร์ต่าง ๆ จากใน Control Panel ของ Main Group ใน Program Manager ของวินโดวส์ ภายในช่องหน้าต่าง Control Panel เราจะเห็นไอคอน ODBC เพื่อเรียกใช้ทำการติดตั้ง ถ้าเราใช้ Excel เวอร์ชัน 5.0 ล่าสุด ไอคอนการติดตั้งไดรเวอร์จะเพิ่มอยู่ในกรุ๊ปของ Microsoft Office เลยทันที ทำให้เราใช้ได้สะดวกยิ่งขึ้น

ODBC Driver Manager จะทำงานต่าง ๆ ดังนี้

- ทำการแม็พแหล่งข้อมูลเข้ากับไดรเวอร์เฉพาะนามสกุล DLL ของแหล่งข้อมูลนั้นตามที่ระบุในไฟล์ ODBC .INI
- เรียกติดตั้งค่าเริ่มต้นของ ODBC
- กำหนดการเรียกเข้าฟังก์ชัน ODBC ให้กับไดรเวอร์ต่าง ๆ
- กำหนดการตรวจความถูกต้องของพารามิเตอร์ที่ใช้ประกอบการเรียก ODBC

ส่วนไดรเวอร์จะเป็นไฟล์ DLL ที่ยึดหรือสร้างตามมาตรฐานฟังก์ชัน ODBC สำหรับการเรียกติดต่อกับแหล่งข้อมูล ซึ่งจะทำหน้าที่เชื่อมต่อแหล่งข้อมูล ส่งการร้องขอกับแหล่งข้อมูล แปลงข้อมูลให้อยู่ในรูปแบบที่ต้องการของโปรแกรมประยุกต์ ส่งข้อมูลผลลัพธ์ไปยังโปรแกรมประยุกต์ จัดข้อผิดพลาดในรูปแบบรหัสและส่งไปยังโปรแกรมประยุกต์ เป็นต้น

ODBC Functions

ฟังก์ชัน ODBC ที่กำหนดโดยบริษัทไมโครซอฟต์ ได้ยึดข้อกำหนด SQL CLI ของ SAG เป็นแกนสำคัญกับได้เพิ่มเติมขยายให้มีมากกว่า กล่าวคือ ฟังก์ชันของ SQL CLI จะมีอยู่เพียง 22 ฟังก์ชัน แต่ ODBC จะมีถึง 51 ฟังก์ชัน ฟังก์ชัน ODBC จะแบ่งได้เป็น 3 ระดับ ระดับแรกเป็นระดับกลางที่คลุมฟังก์ชันจาก SQL CLI ทั้งหมด รวม 22 ฟังก์ชัน ส่วนที่มากกว่าของ ODBC จะจัดอยู่ในระดับ 1 และระดับ 2

ตารางที่ 1 เป็นตารางฟังก์ชันของ ODBC แยกตามประเภทใช้งาน ซึ่งจะบอกชื่อฟังก์ชัน ระดับของฟังก์ชัน และคำอธิบายฟังก์ชัน ฟังก์ชันระดับ 1 และ 2 ที่เกินจาก SQL CLI หลายฟังก์ชันเป็นฟังก์ชันที่มีประโยชน์และจำเป็น คาดว่าในอนาคตจะได้รับการบรรจุเข้าใน SQL CLI ของ SAG ต่อไปค่อนข้างแน่

ตารางที่ 1 ฟังก์ชันของ ODBC แยกตามประเภทของงานโดยคอลัมน์สุดท้ายเป็นข้อมูล Q+E ODBC Pack เครื่องหมายถูกจะหมายถึงฟังก์ชันนั้นมีในทุกไดรเวอร์แล้ว และเครื่องหมายผิดหมายถึงไม่มีฟังก์ชันนี้

การต่อกับแหล่งข้อมูล			
ชื่อฟังก์ชัน	จัดอยู่ในกลุ่ม	คำอธิบาย	Q+E ODBC
SQLAllocEnv	Core	Obtains an environment handle. One environment handle is used for one or more connections.	✓
SQLAllocConnect	Core	Obtains a connection handle.	✓
SQLConnect	Core	Connects to a specific driver by data source name, user ID, and password.	✓
SQLDriverConnect	Level 1	Connects to a specific driver by connection string or requests that the Driver Manager and driver display connection dialogs for the user.	✓
SQLBrowseConnect	Level 2	Returns successive levels of connection attributes and valid attribute values. When a value has been specified for each connection attribute, connects to the data source.	เฉพาะ Informix Driver Ingres Driver, Netware SQL, Oracle, SQL Server และ SQLBase

ตารางที่ 1 (ต่อ)

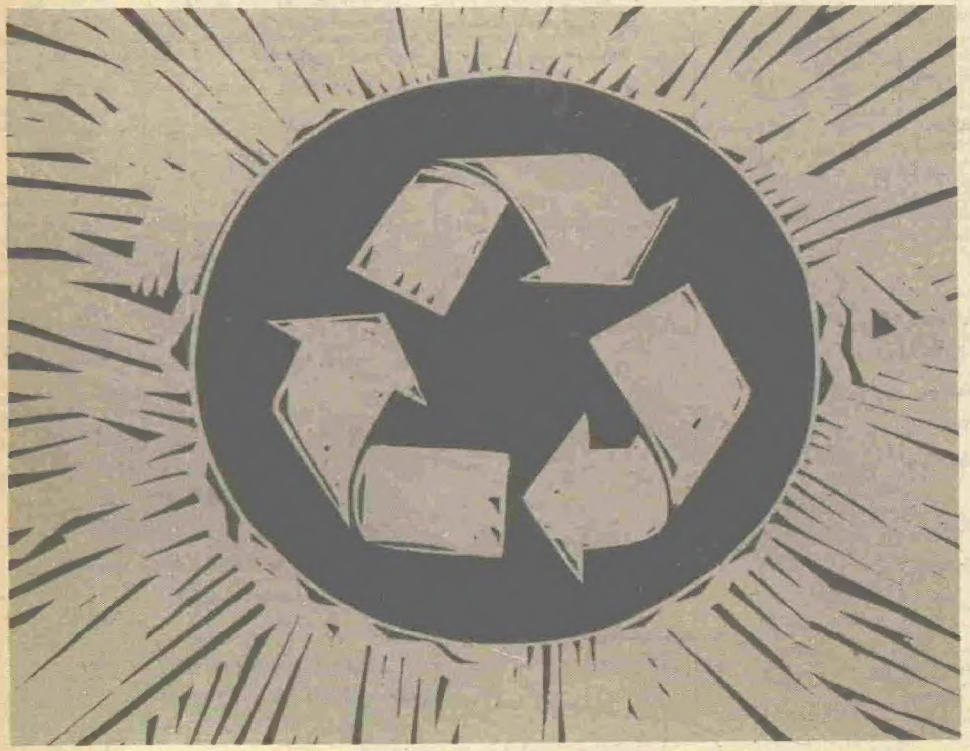
การขอดูข้อมูลเกี่ยวกับไดร์ฟเวอร์และแหล่งข้อมูล			
SQLDataSources	Level 2	Returns the list of available data sources.	✓
SQLGetInfo	Level 1	Returns information about a specific driver and data source.	✓
SQLGetFunctions	Level 1	Returns supported driver functions.	✓
SQLGetTypeInfo	Level 1	Returns information about supported data types.	✓
การตั้งและเรียกดูตัวเลือกของไดร์ฟเวอร์			
SQLSetConnectOption	Level 1	Sets a connection option.	✓
SQLGetConnectOption	Level 1	Returns the value of a connection option.	✓
SQLSetStmtOption	Level 1	Sets a statement option.	✓
SQLGetStmtOption	Level 1	Returns the value of a statement option.	✓
การเตรียมการร้องขอ SQL			
SQLAllocStmt	Core	Allocates a statement handle.	✓
SQLPrepare	Core	Prepares an SQL statement for later execution.	✓
SQLSetParam	Core	Assigns storage for a parameter in an SQL statement.	✓
SQLParamOptions	Level 2	Specifies the use of multiple values for parameters.	✘
SQLGetCursorName	Core	Returns the cursor name associated with a statement handle.	✓
SQLSetCursorName	Core	Specifies a cursor name.	✓
SQLSetScrollOptions	Level 2	Sets options that control cursor behavior.	✓
การส่งการร้องขอ			
SQLExecute	Core	Executes a prepared statement.	✓
SQLExeDirect	Core	Executes as statement	✓
SQLNativeSql	Level 2	Returns the text of an SQL statement as translated by the driver.	✓
SQLDescribeParam	Level 2	Returns the description for a specific parameter in a statement.	✘
SQLNumParams	Level 2	Returns the number of parameters in a statement.	✓
SQLParamData	Level 1	Used in conjunction with SQLPutData to supply parameter data at execution time. (Useful for long data values.)	✓
SQLPutData	Level 1	Send part or all of a data value for a parameter. (Useful for long data values.)	✓

ข้อมูลเรียกขอและข้อมูลเกี่ยวกับผลเรียกขอ			
SQLRowCount	Core	Returns the number of rows affected by an insert, update, or delete request.	✓
SQLNumResultCols	Core	Returns the number of columns in the result set.	✓
SQLDescribeCol	Core	Describes a column in the result set.	✓
SQLColAttributes	Core	Describes attributes of a column in the result set.	✓
SQLBindCol	Core	Assigns storage for a result column and specifies the data type.	✓
SQLFetch	Core	Returns a result row.	✓
SQLExtendedFetch	Level 2	Returns multiple result rows.	✓
SQLGetData	Level 1	Returns part or all of one column of one row of a result set. (Useful for long data values.)	✓
SQLSetPos	Level 2	Positions a cursor within a fetched block of data.	✘
SQLMoreResults	Level 2	Determines whether there are more result sets available and, if so, initializes processing for the next result set.	✓
SQLError	Core	Returns additional error or status information.	✓
การขอดูข้อมูลเกี่ยวกับตารางแคตาล็อกของระบบ			
SQLColumnPrivileges	Level 2	Returns a list of columns and associated privileges for one or more tables.	เฉพาะ SQL Server
SQLColumns	Level 1	Returns the list of column names in specified tables.	✓
SQLForeignKeys	Level 2	Returns a list of column names that comprise foreign keys, if they exist for a specified table.	เฉพาะ SQL Server
SQLPrimaryKeys	Level 2	Returns the list of column name(s) that comprise the primary key for a table.	เฉพาะ SQL Server
SQLProcedureColumns	Level 2	Returns the list of input and output parameters, as well as the columns that make up the result set for the specified procedures.	เฉพาะ SQL Server
SQLProcedures	Level 2	Returns the list of procedure names stored in a specific data source.	เฉพาะ SQL Server
SQLSpecialColumns	Level 1	Returns information about the optimal set of columns that uniquely identifies a row in a specified table, or the columns that are automati-	✓

ตารางที่ 1 (ต่อ)

		cally updated when any value in the row is updated by a transaction.	✓
SQLStatistics	Level 1	Returns statistics about a single table and the list of indexes associated with the table.	
SQLTablePrivileges	Level 2	Returns a list of tables and the privileges associated with each table.	เฉพาะ SQL Server
SQLTables	Level 1	Returns the list of table names stored in a specific data source.	✓
การยกเลิกคำสั่ง			
SQLFreeStmt	Core	Ends statement processing and closes the associated cursor, discards pending results, and, optionally, frees all resources associated with the statement handle.	✓
SQLCancel	Core	Cancels an SQL statement	✓
SQLTransact	Core	Commits or rolls back a transaction.	✓
การยกเลิกการเชื่อมต่อ			
SQLDisconnect	Core	Closes the connection.	✓
SQLFreeConnect	Core	Releases the connection handle.	✓
SQLFreeEnv	Core	Releases the environment handle.	✓

หมายเหตุ Q+E ODBC facel จะมีฟังก์ชัน SQLd ExtendedFetch สำหรับสนับสนุนการดึงข้อมูลย้อนหลังและสุ่มเคา สนับสนุนการเลื่อนเคอร์เซอร์ในชุดข้อมูลไปมาได้ ซึ่งพบในไดร์ฟเวอร์ของ XDB, SQLBase, AllBase, dBASE, Excel, Paradox.

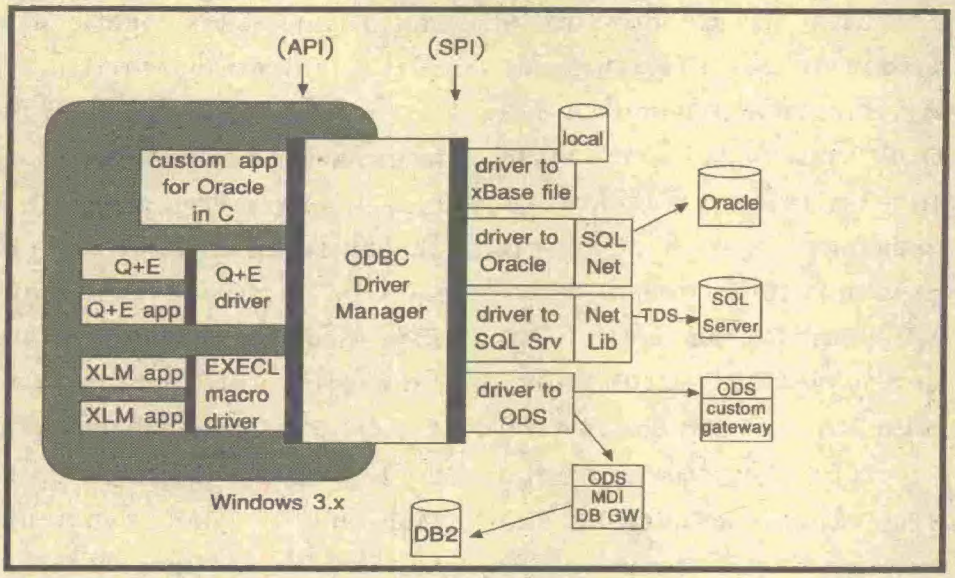


ในปัจจุบันมีบริษัทต่าง ๆ ผลิต ODBC ไดรเวอร์ออกมามากพอสมควร เพื่อให้สามารถเรียกติดต่อ IBM DB2 DEC Rdb, Microsoft SQL Server, Borland Paradox, Oracle, Xbase, Excel ไฟล์ .XLS และไฟล์ข้อความปกติ ตัวอย่างเช่น บริษัท Q+E Software ผลิต ODBC driver Pack ติดต่อแหล่งข้อมูลมากกว่า 20 แหล่ง ดังจะได้กล่าวถึงต่อไป รูปที่ 4 เป็นภาพแสดงตัวอย่างการเรียกติดต่อฐานข้อมูลหลายตระกูลซึ่งในกรณีของการติดต่อกับ IBM DB/2 จะใช้หลักการผสมระหว่าง ODBC กับแบบใช้ช่องประตูร่วมผ่านไดรเวอร์ ODS ของไมโครซอฟต์ไปยังซอฟต์แวร์ช่องประตูของ Micro DecisionWare Inc. (MDI) ในการติดต่อไป DB/2 อีกต่อหนึ่ง

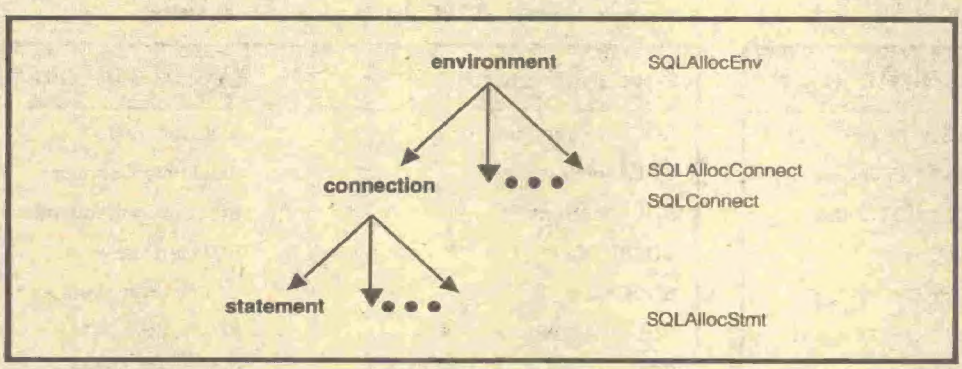
ODBC Programming

เพื่อให้เห็นวิธีการเรียกใช้ ODBC ฟังก์ชันดียิ่งขึ้น ในที่นี้จึงขอเสนอภาพการเรียกใช้ดังรูปที่ 5 ที่โปรแกรมประยุกต์จะต้องกำหนดบอกสภาพแวดล้อมของ ODBC ก่อนด้วยการใช้ฟังก์ชัน SQLAllocEnv จากนั้นจึงกำหนดการเชื่อมต่อด้วย SQLAllocConnect และ SQLConnect ต่อจากนั้นจึงค่อยกำหนดด้วยคำสั่งกับการเชื่อมต่อด้วย SQLAllocStmt ซึ่งรูปแบบจะเรียกตามลำดับจาก environment connection-statement

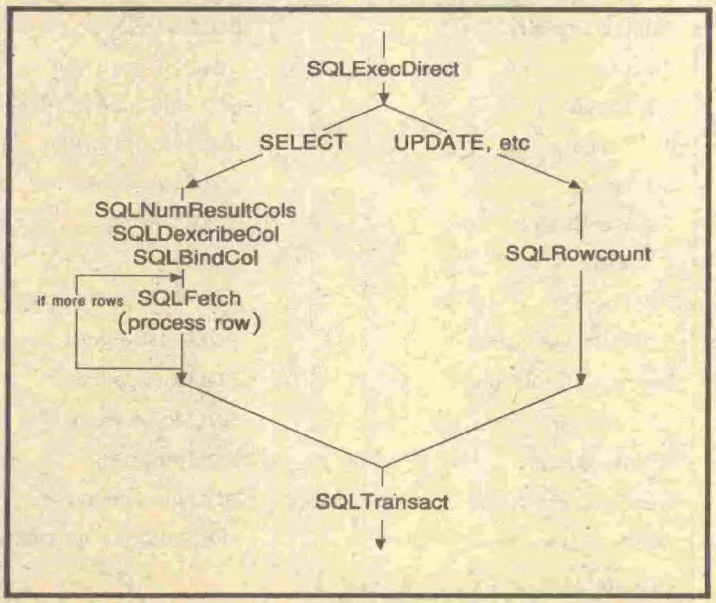
การให้ทำงานตามคำสั่งอาจมีทางเลือก 2 ทางคือ การทำงานเพียงรอบเดียว หรือการทำงานซ้ำหลาย ๆ รอบ โปรแกรมการทำงานรอบเดียวอาจแสดง



รูปที่ 4 ตัวอย่างการเชื่อมต่อและเรียกใช้ข้อมูลหลายตระกูล



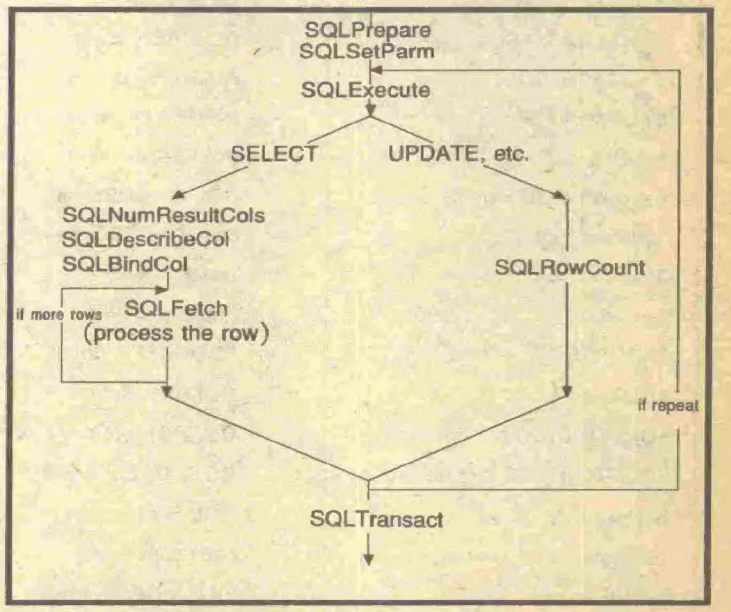
รูปที่ 5 ลำดับการเชื่อมต่อและการเรียกใช้แหล่งข้อมูล



รูปที่ 6 การทำงานรอบเดียวของ ODBC

ได้ดังรูปที่ 6 กรณีนี้โปรแกรมจะเริ่มด้วยฟังก์ชัน SQLExecDirect แล้วเลือกระหว่างคำสั่งเรียกค้น SELECT กับคำสั่งไม่เรียกค้นอื่น ๆ เช่นการอัพเดต ในการเรียกค้นก็จะเริ่มจากจำนวนคอลัมน์ที่ต้องการแสดงด้วย SQL NumResultsCols การอธิบายแต่ละคอลัมน์ด้วย SQLDescribeCol และตัวแปรของค่าแต่ละคอลัมน์ด้วย SQL-BindCol จากนั้นจึงตามด้วยการเรียกทีละแถววนซ้ำ ๆ ด้วย SQLFetch จนครบ SQL Transact

การทำงานหลายรอบซ้ำ ๆ แสดงดังรูปที่ 7 ที่จะต้องมีคำสั่ง SQLPrepare เพื่อการตระเตรียมกระจายและเลือกคำสั่ง SQL ต่าง ๆ คำสั่ง SQL SetParm เมื่อการกำหนดและกำกับพารามิเตอร์กับคำสั่ง SQL และคำสั่ง SQLExecute เพื่อการไปทำงานตามคำสั่ง SQL ต่าง ๆ ต่อไป



รูปที่ 7 การทำงานหลายรอบซ้ำ ๆ ของ ODBC

Q+E ODBC Pack

บริษัท Q+E Software Inc. (เดิมชื่อ Pioneer Software) จากการที่บริษัทไมโครซอฟต์ต้องการให้เอ็กเซลที่เป็นสเปรดชีตของตนสามารถเรียกติดต่อข้อมูลภายนอกอื่น เช่น ORACLE ได้ เพื่อสามารถไปแข่งขันกับโลตัส 1-2-3 ที่มีดาต้าเบสติดต่อกับ ORACLE ได้อยู่แล้ว ไมโครซอฟต์จึงเจรจากับ Q+E Software Inc. เพื่อนำไดรเวอร์ของ Q+E มาใส่ในเอ็กเซล ทำให้ชื่อ Q+E ติดตลาดวงการคอมพิวเตอร์ขึ้นมาได้ทันที

บริษัท Q+E Software ได้รวบรวมไดรเวอร์สำหรับเชื่อมต่อ และเรียกใช้แหล่งข้อมูลภายนอกมารวมเรียก ODBC driver Pack ออกจำหน่าย ในชุดไดรเวอร์นี้สามารถติดต่อข้อมูลแหล่งต่าง ๆ ดังตารางที่ 2 ซึ่งในบางฐานข้อมูลจำเป็นต้องมีช่องประตูของ MDI (Micro Decisionware Inc.) หรือช่องประตูของ DDCS/2 (IBM Distributed Database Connection Service/2)

Q+E ODBC Pack จะสนับสนุนรูปแบบเต็มของฟังก์ชัน ODBC ของไมโครซอฟต์ค่อนข้างจะสมบูรณ์ ดังตารางที่ 3 จะมีเพียงส่วนน้อยที่แตกต่างออกไป ในที่นี้จะแสดงรายละเอียดไว้แล้วในคอลัมน์สุดท้ายของตารางที่ 1

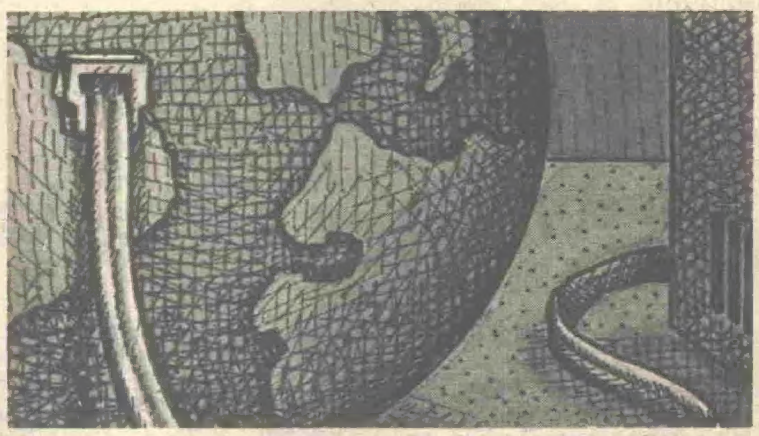
สำหรับในประเทศไทยเอง บริษัท International Software Factory Co., Ltd. โทร. 274-3530, 274-3457-9 เป็นผู้แทนจำหน่ายผลิตภัณฑ์ของ Q+E Software ซึ่งผู้เขียนต้องขอขอบคุณบริษัทที่ได้เอื้อเฟื้อ ODBC pack มาให้ทดลองใช้งานในที่นี้ด้วย ผลิตภัณฑ์ของ Q+E Software ยังมีอยู่อีก เช่น Q+E Database Library, Q+E MultiLink/VB ฯลฯ ซึ่งล้วนเป็นผลิตภัณฑ์น่าใช้งาน ผู้เขียนจะติดตามนำมาเสนอในโอกาสต่อไป

ตารางที่ 2 แหล่งข้อมูลที่เรียกผ่านไดรเวอร์ของ Q+E ได้

ระบบฐานข้อมูล	ชื่อไดร์ฟเวอร์ใน Q+E
Btrieve	Btrieve driver
Clipper	dBASE driver
Database Manager	OS/2 DB2/2 driver
DB2 (MDI gateway)	MDI driver
DB2 (DDCS/2 gateway)	OS/2 DB2/2 driver
DB2/2	OS/2 DB2/2 driver
dBASE II, II, IV files	dBASE driver
Excel .XLS files	Excel driver
HP ALLBASE/SQL	ALLBASE driver
HP IMAGE/SQL	ALLBASE driver
INFORMIX	INFORMIX driver
INGRES	INGRES driver
Microsoft SQL Server	SQL Server driver
NetWare SQL	NetWare SQL driver
Oracle	Oracle driver
Paradox	Paradox driver
PROGRESS	PROGRESS driver
SQLBase	SQLBase driver
SQL/400 (DDCS/2 gateway)	OS/2 DB2/2 driver
SQL/DS (DDCS/2 gateway)	OS/2 DB2/2 driver
Sybase SQL Server	SQL Server driver
Teradata (MDI gateway)	MDI driver
Text files	Text driver
XDB	XDB driver

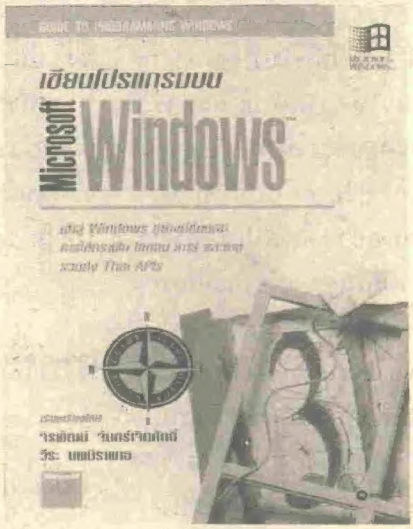
ตารางที่ 3 ฟังก์ชัน ODBC ใน Q+E Database Driver

Core Functions	Level 1 Functions
SQLAllocConnect	SQLColumns
SQLAllocEnv	SQLDriverConnect
SQLAllocStmt	SQLGetConnectOption
SQLBindCol	SQLGetData
SQLCancel	SQLGetFunctions
SQLColAttributes	SQLGetInfo
SQLConnect	SQLGetStmtOption
SQLDescribeCol	SQLGetTypeInfo
SQLDisconnect	SQLParamData
SQLError	SQLPutData
SQLExecDirect	SQLSetConnectOption
SQLExecute	SQLSetStmtOption
SQLFetch	SQLSpecialColumns
SQLFreeConnect	SQLStatistics
SQLFreeEnv	SQLTables
SQLFreeStmt	**Level 2 Functions**
SQLGetCursorName	SQLDataSources
SQLNumResultCols	SQLExtendedFetch
SQLPrepare	SQLMoreResults
SQLRowCount	SQLNativeSql
SQLSetCursorName	SQLNumParams
SQLSetParam	SQLSetScrollOptions
SQLTransact	



สรุป

Open Database Connectivity หรือ ODBC ของไมโครซอฟต์เป็นเทคโนโลยีที่เปิดกว้างให้ผู้ใช้งานสามารถเชื่อมต่อและเรียกใช้ฐานข้อมูลหลายแหล่งหลายตระกูลได้สะดวกสบายยิ่งขึ้น การพัฒนาโปรแกรมประยุกต์ก็จะพัฒนาโดยไม่ต้องมาทำการแก้ไข หากต้องการติดต่อกับแหล่งข้อมูลใดก็เพียงเปลี่ยนไดรเวอร์ของแหล่งข้อมูลนั้น แนวโน้มของ ODBC ของไมโครซอฟต์ได้รับการขานรับจากวงการพอสมควร คือมีสมาชิกในกลุ่ม WOSA (Windows Open Service Architecture) อยู่ระดับหนึ่งช่วยสนับสนุนอยู่ ขณะเดียวกันมีบริษัทซอฟต์แวร์เฮ้าส์อื่น ๆ เช่น Q+E Software Inc. ช่วยพัฒนาซอฟต์แวร์ในลักษณะของไดรเวอร์และไลบรารีมาเสริมอีกแรงหนึ่ง แต่ขณะเดียวกันก็ยังมีอีกค่ายหนึ่งของบริษัท Borland ก็ได้ออก IDAPI ในลักษณะเดียวกัน ODBC ซึ่งมีการผนวก SQL CLI ไว้ด้วย และได้เพิ่มเติมจำนวนฟังก์ชันให้มากกว่าใน ODBC เสียอีก ซึ่งการเพิ่มคู่แข่งขันเช่นนี้ผู้ใช้อย่างเรา ๆ ท่าน ๆ ต่างได้ประโยชน์แน่นอน รายละเอียดของ IDAPI จะได้นำเสนอต่อไป



เขียนโปรแกรมบน
Microsoft Windows™

Windows คือระบบปฏิบัติการหรือ OS สุดฮิตของวันนี้ ด้วยการแสดงผลกราฟิกอันสวยงามน่าทึ่ง ใช้ง่าย สามารถทำงานได้หลายอย่างในเวลาเดียวกัน และใช้หน่วยความจำได้มากกว่า 640K

เปลี่ยนโปรแกรมที่คุณเขียนขึ้นให้ใช้ง่ายด้วยระบบไอคอน เมาส์ และเมนูที่มีพร้อมอยู่แล้วใน Windows พร้อมด้วยความสามารถทางด้านกราฟิกอันน่าทึ่ง การใช้หน่วยความจำได้มากมายถึง 16 เมกะไบต์ และด้วยการเรียนรู้ Windows ทีละเล็กทีละน้อยอย่างเป็นระบบและเป็นขั้นตอน จึงเหมาะสำหรับโปรแกรมเมอร์ที่ต้องการเขียนโปรแกรมบน Windows หรือต้องการย้ายโปรแกรมที่เขียนขึ้นไปทำงานให้ดีกว่าเดิมบน Windows

วางตลาดแล้ว

GUIDE TO PROGRAMMING WINDOWS

วีเอ็ด

หาซื้อได้ตามร้านหนังสือชั้นนำทั่วไป